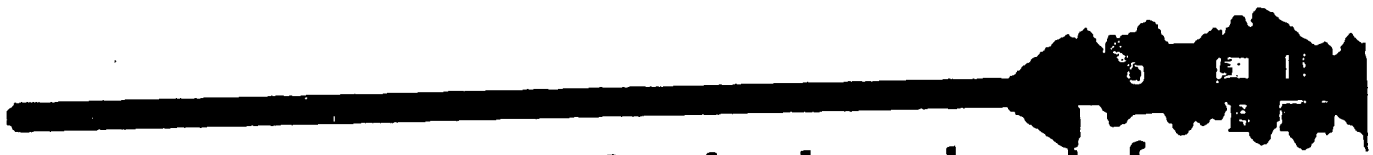
Through-put of Restoration Requests

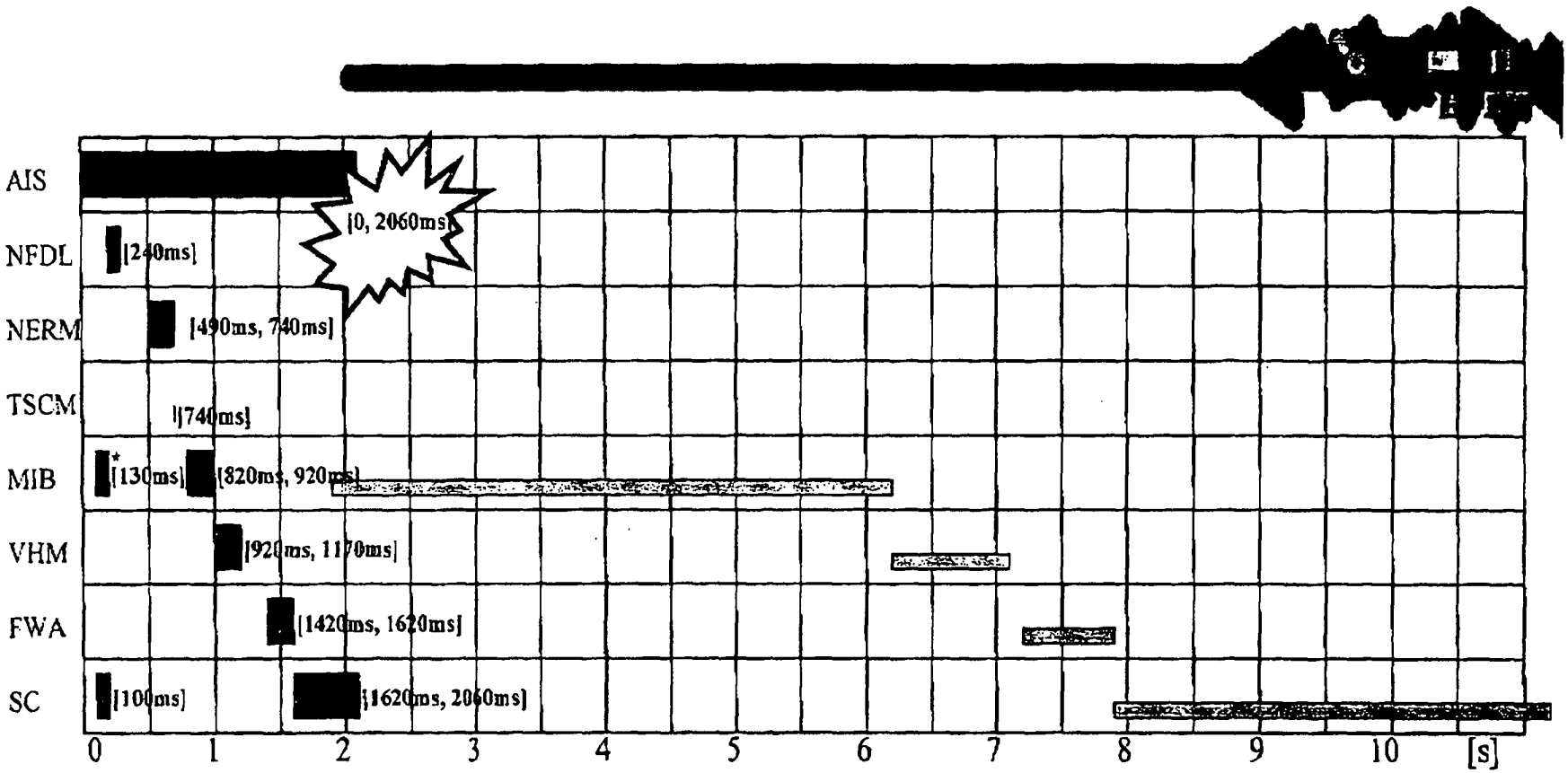
ALCATEL



In future it is necessary to increase also the through-put of rest-oration requests per time unit, e.g. a 2nd request is early accepted by MIB

- MIB has to abort current consolidation. Both consolidation requests have to be merged to a single one
- VHM consolidation is not interruptible
- SC: early availability after 'Fetch-ahead-StartUpd'
- Overload defense: Ensure that consolidation isn't permanently blocked (risk to loose non-consolidated configuration)
- SC: minimized dead time during processing of consolidation requests

Improvement



= Signal

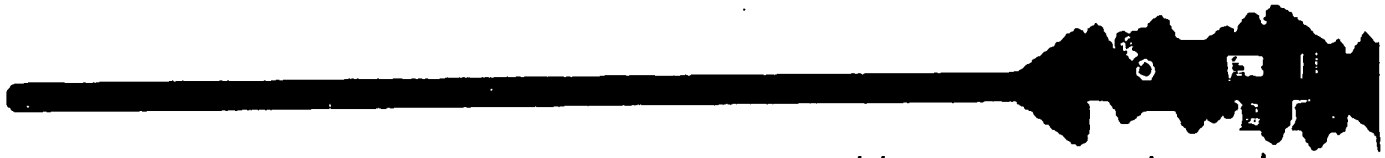
= improved processing / Fetch-Ahead

= Consolidation, delayed MIB processing to avoid parallel activity MIB/VHM/FW, time approximately as today

Fig. 11

Improvement Summary

ALCATEL



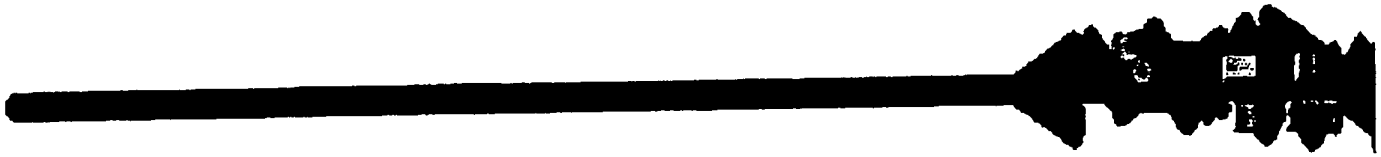
	old	estimated
◆ faster polling of primary alarms	410ms	=> 100ms
◆ reduce hold off in MIB	310ms	=> 110ms
◆ cut NP hold off to 200ms after 1st prim. alarm	830ms	=> 250ms

Two pass approach in the NE, Fetch-Ahead "fast & less secure" and Consolidation "slow & secure":

◆ MIB (only filtering & message conversion)	4300ms	=> 100ms
◆ no persistency in VHM, reduced cfg. data	750ms	=> 200ms
◆ FWA receives/processes reduced data	690ms	=> 200ms
◆ Connection focused ASIC provisioning	2710ms	=> 500ms
	10000ms	=>1460ms

Improvement: 8540ms, signal fails for 10600ms-8540ms = **2060ms**

Path Setup FW



- Up to now no differentiation Fetch-Ahead / Consolidation foreseen
- Is CS configuration getting the new bottleneck ?
  - It is necessary to analyze if configuration of the CS has to be done in another way, e.g. is board reload still optimal for such big matrices ?
  - Needs connectivity a higher priority during restoration requests ?

ALCATEL

◆ Fetch-ahead

- SC starts ASIC provisioning limited to the creation of the new cross connections:
  - calculation of ASIC maps (minimized difference to actual ASIC map) for matrix, unequipped generators, concatenation (automode ?)
  - delta-writing of ASIC maps to matrices first and where necessary also to port ASICs / link boards
  - take advantage of knowing that there is no MSP/ MSSPRING / Port EPS during restoration

2700ms → 500ms

◆ Consolidation

- complete configuration (normal way)

Path Setup FWA

ALCATEL

- **Fetch-ahead**
  - FWA: prepare & send only blocks for concatenation & connectivity (MSA, HO-Connectivity, ...)
  - Communication limited to active SC
  - Send a Fetch-ahead-UpdProv to the SC.

690ms → 200ms

- **Consolidation**
  - as today

Path Setup VHM

ALCATEL

- **Fetch-ahead**
  - only connection relevant configuration is processed
  - without persistency
  - VHM forwards 'Fetch-ahead indication' via FWA to SC and sends only connection relevant information to FWA
  - between Fetch-ahead and consolidation no alarm reporting
  - optimized reconfiguration of Center Stage
  - container message to FWA ?

750ms → 200ms

- **Consolidation**
  - as today (persistency), partly rewriting Fetch-ahead configuration
  - reuse results from Fetch-ahead: routing (!)
  - afterwards alarm handling available again

Path Setup / MIB

Restoration requests from NP are parsed twice in the MIB:

- **Fetch-ahead**
  - filter for connection relevant information: concatenation, connectivity
  - translate filtered data into VHM configuration request
  - no MO creation / deletion, no persistency

  => earliest hand over to VHM

  4300ms → 100ms

- **Consolidation**
  - creation / deletion of monitors / MOs / persistency
  - creation of spare link supervision

Abstract

Fast Restoration

A method for re-configuring a network element of a transmission network to restore traffic after occurrence of a failure is proposed. Each configuration request is divided into two phases: At first the "Fetch-Ahead" phase and then the "Consolidation". During Fetch-Ahead no consistency checks are performed, the changes are not made persistent in the database and only the absolutely necessary changes are done. After Fetch-Ahead these changes are temporarily valid in all SW layers. During Consolidation all the remaining things, which are skipped during Fetch-Ahead, are done.

(Figure 2)

What is claimed is

1. A method of re-configuring a network element of a transmission network to restore traffic after a failure, said method comprising the steps of:

- generating a configuration request to switch an alternative connection through said network element,
- performing said request in a first fetch-ahead phase, thereby switching said requested connection;
- performing in a second phase a consolidation of said request, said consolidation comprising a consistency check of said request and storing of the configuration changes in a persistent local database.

2. Network element with layered control software for performing said two phase re-configuration.

ALCATEL

Path Setup in NE

- **Fetch-ahead**
  - processing limited to connection relevant information
  - earliest hand over to next layer
  - MIB / VHM / SC may enter a transient configuration but with stable connections, e.g. even without consolidation the connections remain stable
  - no MIB / VHM persistency
  - no NAU redundancy
  - in case of restarts only the Fetch-ahead step may fail
- **Consolidation**
  - complete configuration, security as today (persistency)
  - configuration from Fetch-ahead is partly re-written

Two pass processing in NE SW layers

- first pass **'Fetch-ahead'**
  - ignore all that is not restoring traffic or that is done for security

- second pass **'Consolidation'**
  - provide all missing configuration, persistency etc.

Alarm Detection / Path Calculation

- Faster polling for primary alarms

  first alarm at 410ms reported from SC

  polling today >=500ms but 100ms for max. 128 STM interfaces per shelf realistic (?)

  410ms → 100ms

- Reduce MIB hold off for alarm collection

  first alarm detected from SC at 410ms, received by MIB at 420ms, received from NP: 740ms

  310ms → 150ms

- Reduce NP hold off for alarm collection

  first alarm at 740ms, protection path calculated at 1560ms

  (Cut remaining hold off to 200ms after reception of a primary alarm ?)

  830ms → 250ms

Consequences

To achieve 2 seconds restoration time each single component has to be optimized (every millisecond counts)

- alarm detection
- path calculation
- path implementation
- communication

Options:

- faster alarm detection
- slowing down less time-critical features
- reduce security (inhibit persistency)
- by-pass time consuming layers

ALCATEL

Improvements

ALCATEL

Reasons for low Performance

- slow alarm polling
- hold off time in MIB
- hold off time in NP
- creation of MOs in MIB
- persistency in VHM
- communication between FWA - SC
- non optimized ASIC provisioning sequence

Actual Figures LG 1.0 (3 nodes, 8x AU4-4c)

ALCATEL

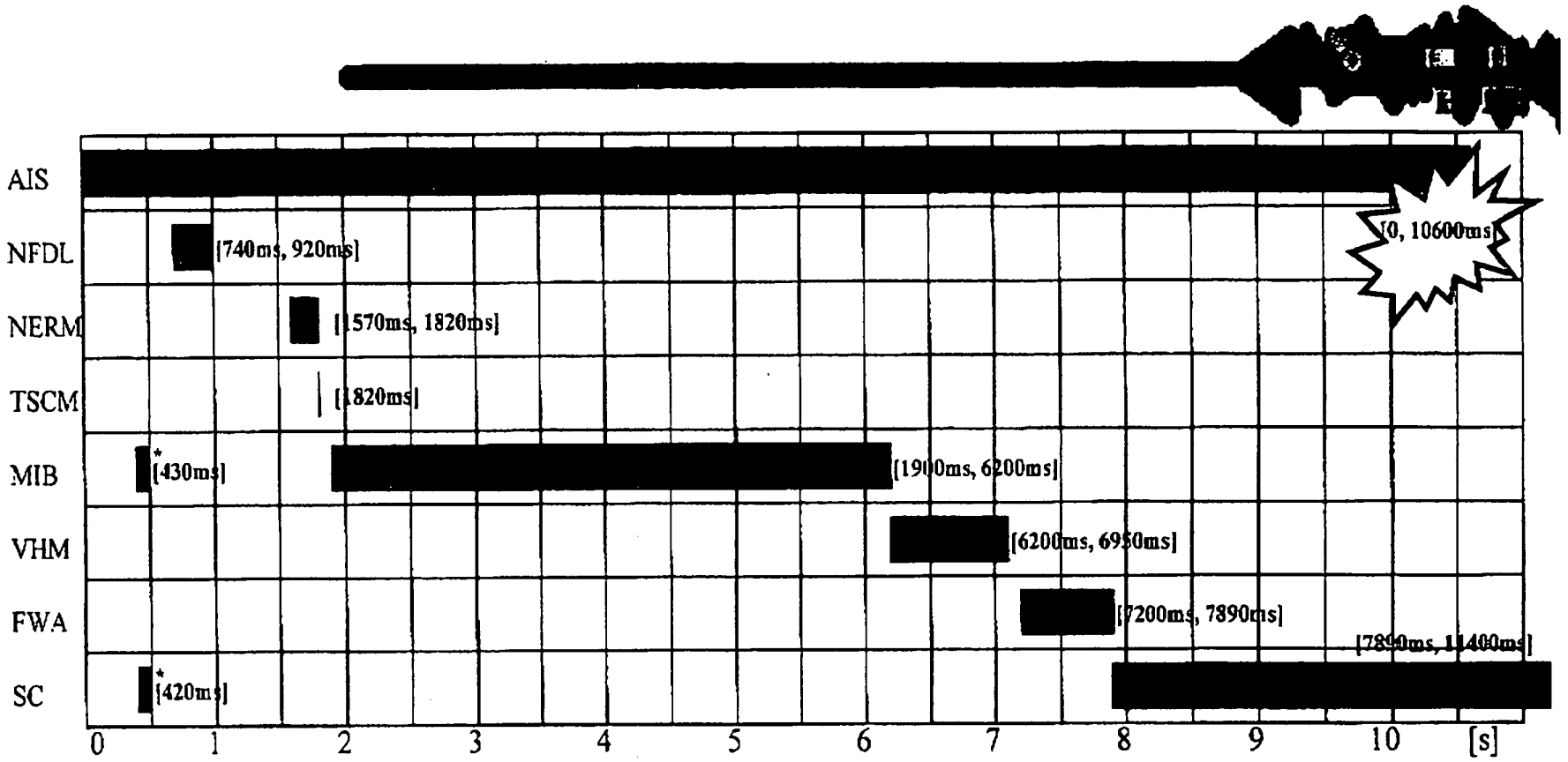


Fig. 8

- **Activity starts when all necessary information is available**
- **Activity ends when all necessary information is passed to the next layer.**
- *** first alarm from network**

Data provided by F.Comi, 03/05/02

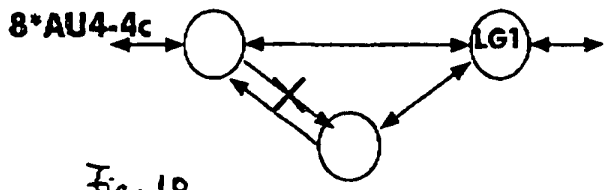


Fig. 10

Test Scenario

- Comparison of protection times is done with a simplified scenario:
  - single point of failure for m x AU4-nc
  - primary fault LOS - easy to locate (no AIS along a path)
  - empty matrices (no rearrangement)
  - simple net topology (few hops -> optimal for distributed rest.)
  - network has no history (only few nominal routes -> optimal for distributed rest.)
  - traffic is restored within the first run (restoration doesn't fail)
- The system performance for the straight forward scenario has to be improved or a different mechanism has to be used (m:n SNCP)

Crucial:

Protection time must be ~ 2 seconds

Minor or not perceived:

- optimal usage of the network resources (convergence to nominal routes)
- down time until the next protection can be executed
- stability in case of double failures
- internal supervision
- supervision of spare links
- behavior when first restoration fails
- protection time when no primary fault detected

37

Actual Situation

ALCATEL

RDI	Remote Defect Indication	285
SAD	Software Architecture Document	286
SC	Shelf Controller	287
SDD	Software Design Document	288
SNCP	Subnetwork Connection Protection	289
SoW	Statement of Work	290
SoW	Statement of Work	291
SPRC	Shared Protection Restoration Combined	292
SUT	Supervisory Unequipped Generator	293
TC	Tandem Connection	294
TSCM	Terminal Station Command Handler	295
TIM	Trail Identifier Mismatch	296
TTP	Trail Termination Point	297
TT	Trail Termination	298
UNEQ	Unequipped	299
VHM	Virtual Hardware Manager	300

ALCATEL

Glossary

This section provides an explanation of those technical terms used specifically in this document. Terms defined in the FTS, SAD or SDD do not have to be listed here.

feature Product Feature or System Enhancement

requirements System Requirements

customer Customer, Customer Surrogate (e.g. Product Management), or user of the system

List of Abbreviations and Acronyms

This section provides the abbreviations and acronyms used specifically in this document. Abbreviations defined in the FTS, SAD or SDD do not have to be listed here.

AIS	Alarm Indication Signal
BSW	Board Software
CCU	CLock and Control Unit
CTP	Connection Termination Point
DXC	Digital Cross Connect
EPS	Equipement Protection Switch
FDS	Feature Design Specification
FSD	Feature Scope Description
FTS	Functional Technical Specification
FW	Firmware
FWA	Firmware Adaptor
IM	Information Model
LG	Lambda Gate
LOS	Loss Of Signal
MIB	Management Information Base
MO	Managed Object
MSD	Message Set Document
MSP	Multiplex Section Protection
NERM	Network Emergency Reconfiguration Manager
NFDL	Network Fault Detection and Location
POH	Path Overhead

CHAPTER 8: Caveats

This chapter describes all of the assumptions, limitations and risks associated with the requirements, architecture and high-level design decisions previously described. Any undesirable behaviors, operational deficiencies, implementation risks, performance issues, functional limitations, and anything else potentially embarrassing is described. 254

8.1 Assumptions

none 255

8.2 Limitations

none 256

8.3 Risks

none 257

8.4 General Notes

none 258

ALCATEL

deactivation of the SUT generator 246

7.1.9 FW Impacts

managing the fetch–ahead flag 247

connection request established in a fast manner by bypassing normal mechanisms (data-base, etc.) 248

managing the consolidation flag 249

consolidation supervision timer and reload handling 250

acceptance of a consolidation request without fetch–ahead 251

- MessageSet updates 252

7.1.10 BSW Impacts

verify if the performance of BSW functions can be improved 253

startHoldOff during fetch–ahead and consolidation 218

- storing and managing concatenation information at the endstage board for the complete shelf 219
- skip I/O board configuration during fetch–ahead (concatenation information stored at ES matrix board) 220
- managing of the fetch–ahead socket 221

general improvements, purify / quantify 222

center stage matrix configuration message built once and duplicated to both matrix copies during fetch–ahead (SDH) 223

messageSet updates 224

- accept implicit concatenation requests 225

repeat connection configuration during consolidation 226

improve processing of alarms 227

7.1.7 FWA Impacts

special mapping for the fetch–ahead container 228

- indication about fetch–ahead to SC and to FW (with message header) 229

startHoldOff during fetch–ahead and consolidation 230

inhibiting PM history handling after startHoldOff message from VHM. History handling is continued after configurable timer (5...30 seconds, in configuration file) 231

- general improvements, purify / quantify 232
- message set update (fetch–ahead and consolidation flag) 233

message set update: concatenation information for a complete shelf 234

fetch–ahead configuration only to active SC and active CCU 235

MessageSet updates 236

7.1.8 SC Impacts

- managing the consolidation flag 237

consolidation supervision timer and reset handling 238

- acceptance of a consolidation request without fetch–ahead 239
- improved alarm polling and reporting 240
- improved ASIC provisioning 241

disabling of I0/I1 OTIF supervision consequent action AIS upon reception of startHoldOff 242

MessageSet updates 243

increasing the maximum message size 244

- general improvements, purify / quantify 245

3A

ALCATEL

CHAPTER 7: Impacts on Segments

This chapter describes the allocation of the requirements to segments. 203

7.1 Impacts

7.1.1 Hardware Impacts

none 204

7.1.2 Basic Framework

none 205

7.1.3 SH/CT Impacts

none 206

7.1.4 MIB Framework Impacts

NP request have to be prioritized 207

7.1.5 MIB Impacts

hold off for alarm collection (configurable via cfg file 0....300ms) 208

- skip hold off for primary faults like LOS and section alarms MS−AIS 209

send primary faults directly to NP before making them persistent 210

translation and filtering of the restoration request 211

consolidation is started upon the reception of the fetch−ahead response 212

- indication about fetch−ahead to VHM 213
- general improvements, purify / quantify, LogRecords 214

7.1.6 VHM Impacts

inhibiting persistency management between fetch−ahead and consolidation 215

- send an indication about fetch−ahead to FWA 216

send an indication about consolidation to FWA (i.e. the former Restoration indication) 217

6.2 Behavior between fetch–ahead and c nsolidation

During fetch–ahead only MSA (for concatenation), HO TX/RX connection map are sent to the SC. This leads to an inconsistent configuration. To keep the impacts low the SC will not resume the tasks between fetch–ahead and consolidation, i.e. alarm task, event reporter will not report new alarms. [198]

In case an equipment failure occurs (internal fiber cut, board extracted etc.) the problem is not recovered because EPS, etc. is blocked with the suspended tasks. [199]

Between fetch–ahead and consolidation misconnections are not detected and not forced to AIS. [200]

The PM counting is continuing since it is done in a dedicated hardware device. [201]

When the tasks are resumed with the consolidation process they start reporting the alarm differences in comparison to the status before fetch–ahead. [202]

- MIB rejects restoration request during consolidation 189
  MIB will close the restoration socket and VHM interprets this as a MIB restart, see above.

6.1.2 VHM

- restart before the fetch–ahead data has been completely sent to SC, CCU: 190
  VHM 'forgets' the incomplete processed fetch–ahead configuration and some paths may be already restored and others not because some SCs / CCUs may have received fetch–ahead configuration and others not. As soon as MIB detects a VHM restart it starts immediately the consolidation (including the consolidation tag in the messages) and the fetch–ahead is not repeated. If the restart was fast enough then the consolidation arrives before the consolidation supervision timers at SC and CCU level expire and for these processing elements it will be like a normal consolidation. The other SCs and CCUs which didn't receive the fetch–ahead shall accept the consolidation configuration as a normal configuration request.
- restart between fetch–ahead and before having sent all consolidation to SC / CCU: 191
  the paths are already restored due to the complete execution of the fetch–ahead. The consolidation request coming from MIB is not completely processed inside VHM and MIB will repeat the consolidation request after a time out. It may happen that the consolidation arrives too late at SC / CCU level and that they've already made a restart. In this case the path is failing again until consolidation arrives.
- Normal configuration request between fetch–ahead and consolidation 192
  VHM shall execute a restart in order to forget the fetch–ahead configuration because the consolidation will not arrive anymore. Also all CCUs / SCs which have already the fetch–ahead will execute a restart when their timer expires or when a normal configuration is received.

6.1.3 FWA Restart

In all situations the actual mechanism applies. This means VHM repeats the configuration after time–out. As a consequence the fetch–ahead may fail and paths are not restored in a fast manner. 193

6.1.4 FW Restart

In case a FW restart is executed before the fetch–ahead is completed the path restoration is delayed until the restart is executed and the consolidation arrives. 194

The restarted FW shall accept a consolidation configuration request even if it doesn't know the related fetch–ahead. 195

6.1.5 SC Restarts / SC EPS

If the restart / EPS happens while the fetch–ahead is processed no fast restoration is achieved. The situation is recovered as soon as the consolidation is executed. 196

The restarted SC shall accept a consolidation configuration request even if it doesn't know the related fetch–ahead. 197

ALCATEL

CHAPTER 6: Robustness

This chapter is introduced to analyse if the new concept is robust against exceptional events. 176

The following exceptional situations have been identified as important for the fetch–ahead / consolidation approach: 177

a) a process crashes or processing element EPS during fetch–ahead happens 178

b) a process crashes or processing element EPS during consolidation 179

c) processing element not reachable 180

d) unexpected message, e.g. a normal configuration request between fetch–ahead and consolidation 181

e) invalid message (access to forbidden resources at MIB level, etc.) 182

Any exceptional situation may defeat the fast restoration which is already or which should be achieved with the fetch–ahead. The path availability may also toggle, e.g. first it is recovered and then it returns to the failing state until consolidation is done. Independent from the success of the fetch–ahead it is most important that the following consolidation is always clearing the situation or that the NE itself falls back to the last known consistent configuration. 183

If an exceptional situation occurred the fetch–ahead is not repeated because the main goal of the fetch–ahead cannot be reached anymore, e.g. because the restart of a segment needs several seconds. 184

6.1 Exceptional Situations

6.1.1 MIB

- restart before the fetch–ahead data has been sent to VHM: 185
  The restoration request will completely fail. NP has to execute a second restoration run.
- restart between fetch–ahead and before sending consolidation to VHM 186
  MIB 'forgets' the restoration request and that the fetch–ahead has been (partly or completely) implemented in all following layers.
  VHM detects that MIB crashed and executes a restart or a rollback to the configuration before fetch–ahead.
  In case the fetch–ahead configuration has been already passed to the SC / CCU it will execute a warmstart using the old database configuration after time–out or when VHM sends a normal configuration request (no fetch–ahead, no consolidation).
- restart after fetch–ahead and after processing of consolidation but before the consolidation is sent to VHM: 187
  The consolidation will be sent as a normal configuration requests. This leads to an SC warmstart / CCU reload. The paths created during fetch–ahead are interrupted until the normal configuration is received.
- 2ndrestoration request from NP before consolidtaion of tch 1st is closed: 188
  MIB may send a 2nd fetch–ahead before VHM has received the consolidtaion of the 1st. Afterwards MIB will send a merged consolidation for both fetch–aheads.

27

ALCATEL

- **Equipment protection (port EPS)** 174
  The NE SW layers have to consider this.
- **Parallel management RM/NP** 175
  Restoration request and RM configuration requests may cause a delay to each other.